The Art of Computer Programming (sometimes known by its initials TAOCP) is a comprehensive monograph written by Donald Knuth that covers many kinds of programming algorithms and their analysis.

Knuth began the project, originally conceived as a single book with twelve chapters, in 1962. The first three volumes of what was then expected to be a seven-volume set were published in 1968, 1969, and 1973. The first installment of Volume 4 (a paperback fascicle) was published in 2005. The hardback Volume 4A, combining Volume 4, Fascicles 0–4, was published in 2011. Additional fascicle installments are planned for release approximately biannually; Volume 4, Fascicle 6 (“Satisfiability”) was released in December 2015.

1 History

After winning a Westinghouse Talent Search scholarship, Knuth enrolled at the Case Institute of Technology (now Case Western Reserve University), where his performance was so outstanding that the faculty voted to award him a master of science upon his completion of the baccalaureate degree. During his summer vacations, Knuth was hired by Burroughs to write compilers, earning more in his summer months than full professors did for an entire year. Such exploits made Knuth a topic of discussion among the mathematics department, which included Richard S. Varga.

Knuth started to write a book about compiler design in 1962, and soon realized that the scope of the book needed to be much larger. In June 1965, Knuth finished the first draft of what was originally planned to be a single volume of twelve chapters. His hand-written first-draft manuscript (completed in 1966) was 3,000 pages long: he had assumed that about five hand-written pages would translate into one printed page, but his publisher said instead that about 1½ hand-written pages translated to one printed page. This meant the book would be approximately 2,000 pages in length. The publisher was nervous about accepting such a project from a graduate student. At this point, Knuth received support from Richard S. Varga, who was the scientific advisor to the publisher. Varga was visiting Olga Taussky-Todd and John Todd at Caltech. With Varga’s enthusiastic endorsement, the publisher accepted Knuth’s expanded plans. In its expanded version, the book would be published in seven volumes, each with just one or two chapters. Due to the growth in the material, the plan for Volume 4 has since expanded to include Volumes 4A, 4B, 4C, 4D, and possibly more.

In 1976, Knuth prepared a second edition of Volume 2, requiring it to be typeset again, but the style of type used in the first edition (called hot type) was no longer available. In 1977, he decided to spend some time creating something more suitable. Eight years later, he returned with TEX, which is currently used for all volumes. The offer of a so-called Knuth reward check worth “one hexadecimal dollar” (100HEX base 16 cents, in decimal, is $2.56) for any errors found, and the correction of these errors in subsequent printings, has contributed to the highly polished and still-authoritative nature of the work, long after its first publication. Another characteristic of the volumes is the variation in the difficulty of the exercises. The level of difficulty ranges from “warm-up” exercises to unsolved research problems. Knuth’s dedication reads:

This series of books is affectionately dedicated to the Type 650 computer once installed at Case Institute of Technology, with whom I have spent many pleasant evenings.
2 Assembly language in the book

All examples in the books use a language called “MIX assembly language”, which runs on the hypothetical MIX computer. Currently, the MIX computer is being replaced by the MMIX computer, which is a RISC version. Software such as GNU MDK exists to provide emulation of the MIX architecture. Knuth considers the use of assembly language necessary for the speed and memory usage of algorithms to be judged.

3 Critical response

*American Scientist* has included this work among “100 or so Books that shaped a Century of Science”, referring to the twentieth century,[3] and within the computer science community it is regarded as the first and still the best comprehensive treatment of its subject. Covers of the third edition of Volume 1 quote Bill Gates as saying, “If you think you’re a really good programmer... read (Knuth’s) *Art of Computer Programming*... You should definitely send me a résumé if you can read the whole thing.”[4] *The New York Times* referred to it as “the profession’s defining treatise”. [5]

4 Volumes

- Volume 1 – Fundamental Algorithms
  - Chapter 1 – Basic concepts
    - 1.1. Algorithms
    - 1.2. Mathematical Preliminaries
      - 1.2.1. Mathematical Induction
      - 1.2.2. Numbers, Powers, and Logarithms
      - 1.2.3. Sums and Products
      - 1.2.4. Integer Functions and Elementary Number Theory
      - 1.2.5. Permutations and Factorials
      - 1.2.6. Binomial Coefficients
      - 1.2.7. Harmonic Numbers
      - 1.2.8. Fibonacci Numbers
      - 1.2.9. Generating Functions
      - 1.2.10. Analysis of an Algorithm
      - 1.2.11. Asymptotic Representations
        - 1.2.11.1. The O-notation
        - 1.2.11.2. Euler’s summation formula
        - 1.2.11.3. Some asymptotic calculations
  - 1.3 MMIX (MIX in the hardback copy but updated by fascicle 1)
    - 1.3.1. Description of MMIX
    - 1.3.2. The MMIX Assembly Language
    - 1.3.3. Applications to Permutations
  - 1.4. Some Fundamental Programming Techniques
    - 1.4.1. Subroutines
    - 1.4.2. Coroutines
    - 1.4.3. Interpretive Routines
      - 1.4.3.1. A MIX simulator
      - 1.4.3.2. Trace routines
    - 1.4.4. Input and Output
    - 1.4.5. History and Bibliography
  - Chapter 10 – Parsing techniques
- Volume 2 – Seminumerical Algorithms
  - Chapter 3 – Random numbers
  - Chapter 4 – Arithmetic
- Volume 3 – Sorting and Searching
  - Chapter 5 – Sorting
  - Chapter 6 – Searching
  - Chapter 7 – Combinatorial searching
  - Chapter 8 – Recursion
- Volume 4 – Combinatorial Algorithms (chapters 7 and 8 released in several subvolumes)
  - Chapter 7 – Combinatorial searching
  - Chapter 8 – Recursion
- Volume 5 – Syntactic Algorithms (as of 2015, estimated for release in 2025)
  - Chapter 9 – Lexical scanning (also includes string search and data compression)

- Volume 6 – The Theory of Context-Free Languages (planned)
- Volume 7 – Compiler Techniques (planned)

4.1 Chapter outline of published volumes

- Volume 1 – Fundamental Algorithms
  - Chapter 1 – Basic concepts
    - 1.1. Algorithms
    - 1.2. Mathematical Preliminaries
      - 1.2.1. Mathematical Induction
      - 1.2.2. Numbers, Powers, and Logarithms
      - 1.2.3. Sums and Products
      - 1.2.4. Integer Functions and Elementary Number Theory
      - 1.2.5. Permutations and Factorials
      - 1.2.6. Binomial Coefficients
      - 1.2.7. Harmonic Numbers
      - 1.2.8. Fibonacci Numbers
      - 1.2.9. Generating Functions
      - 1.2.10. Analysis of an Algorithm
      - 1.2.11. Asymptotic Representations
        - 1.2.11.1. The O-notation
        - 1.2.11.2. Euler’s summation formula
        - 1.2.11.3. Some asymptotic calculations
      - 1.3 MMIX (MIX in the hardback copy but updated by fascicle 1)
        - 1.3.1. Description of MMIX
        - 1.3.2. The MMIX Assembly Language
        - 1.3.3. Applications to Permutations
      - 1.4. Some Fundamental Programming Techniques
        - 1.4.1. Subroutines
        - 1.4.2. Coroutines
        - 1.4.3. Interpretive Routines
          - 1.4.3.1. A MIX simulator
          - 1.4.3.2. Trace routines
        - 1.4.4. Input and Output
        - 1.4.5. History and Bibliography
  - Chapter 10 – Parsing techniques
- Volume 2 – Seminumerical Algorithms
  - Chapter 3 – Random numbers
  - Chapter 4 – Arithmetic
- Volume 3 – Sorting and Searching
  - Chapter 5 – Sorting
  - Chapter 6 – Searching
- Volume 4 – Combinatorial Algorithms (chapters 7 and 8 released in several subvolumes)
  - Chapter 7 – Combinatorial searching
  - Chapter 8 – Recursion
- Volume 5 – Syntactic Algorithms (as of 2015, estimated for release in 2025)
  - Chapter 9 – Lexical scanning (also includes string search and data compression)
4.1 Chapter outline of published volumes

- 2.2.3. Linked Allocation
- 2.2.4. Circular Lists
- 2.2.5. Doubly Linked Lists
- 2.2.6. Arrays and Orthogonal Lists

- 2.3. Trees
  - 2.3.1. Traversing Binary Trees
  - 2.3.2. Binary Tree Representation of Trees
  - 2.3.3. Other Representations of Trees
  - 2.3.4. Basic Mathematical Properties of Trees
    - 2.3.4.1. Free trees
    - 2.3.4.2. Oriented trees
    - 2.3.4.3. The “infinity lemma”
    - 2.3.4.4. Enumeration of trees
    - 2.3.4.5. Path length
    - 2.3.4.6. History and bibliography
- 2.3.5. Lists and Garbage Collection

- 2.4. Multilinked Structures
- 2.5. Dynamic Storage Allocation
- 2.6. History and Bibliography

- Volume 2 – Seminumerical Algorithms
  - Chapter 3 – Random Numbers
    - 3.1. Introduction
    - 3.2. Generating Uniform Random Numbers
      - 3.2.1. The Linear Congruential Method
        - 3.2.1.1. Choice of modulus
        - 3.2.1.2. Choice of multiplier
        - 3.2.1.3. Potency
      - 3.2.2. Other Methods
    - 3.3. Statistical Tests
      - 3.3.1. General Test Procedures for Studying Random Data
      - 3.3.2. Empirical Tests
      - 3.3.3. Theoretical Tests
      - 3.3.4. The Spectral Test
    - 3.4. Other Types of Random Quantities
      - 3.4.1. Numerical Distributions
      - 3.4.2. Random Sampling and Shuffling
    - 3.5. What Is a Random Sequence?
    - 3.6. Summary
  - Chapter 4 – Arithmetic
    - 4.1. Positional Number Systems
    - 4.2. Floating Point Arithmetic
      - 4.2.1. Single-Precision Calculations
      - 4.2.2. Accuracy of Floating Point Arithmetic
    - 4.2.3. Double-Precision Calculations
    - 4.2.4. Distribution of Floating Point Numbers
  - 4.3. Multiple Precision Arithmetic
    - 4.3.1. The Classical Algorithms
    - 4.3.2. Modular Arithmetic
    - 4.3.3. How Fast Can We Multiply?
  - 4.4. Radix Conversion
  - 4.5. Rational Arithmetic
    - 4.5.1. Fractions
    - 4.5.2. The Greatest Common Divisor
    - 4.5.3. Analysis of Euclid’s Algorithm
    - 4.5.4. Factoring into Primes
  - 4.6. Polynomial Arithmetic
    - 4.6.1. Division of Polynomials
    - 4.6.2. Factorization of Polynomials
    - 4.6.3. Evaluation of Powers
    - 4.6.4. Evaluation of Polynomials
  - 4.7. Manipulation of Power Series

- Volume 3 – Sorting and Searching
  - Chapter 5 – Sorting
    - 5.1. Combinatorial Properties of Permutations
      - 5.1.1. Inversions
      - 5.1.2. Permutations of a Multiset
      - 5.1.3. Runs
    - 5.1.4. Tableux and Involutions
    - 5.2. Internal sorting
      - 5.2.1. Sorting by Insertion
      - 5.2.2. Sorting by Exchanging
      - 5.2.3. Sorting by Selection
      - 5.2.4. Sorting by Merging
      - 5.2.5. Sorting by Distribution
    - 5.3. Optimum Sorting
      - 5.3.1. Minimum-Comparison Sorting
      - 5.3.2. Minimum-Comparison Merging
      - 5.3.3. Minimum-Comparison Selection
      - 5.3.4. Networks for Sorting
    - 5.4. External Sorting
      - 5.4.1. Multiway Merging and Replacement Selection
      - 5.4.2. The Polyphase Merge
      - 5.4.3. The Cascade Merge
      - 5.4.4. Reading Tape Backwards
      - 5.4.5. The Oscillating Sort
5 English editions

5.1 Current editions

These are the current editions in order by volume number:

- **Volume 1, Fascicle 1**: *MMIX – A RISC Computer for the New Millennium*. (Addison-Wesley, February 14, 2005) ISBN 0-201-85392-2 (will be in the fourth edition of volume 1)
5.2 Previous editions

5.2.1 Complete editions

These volumes were superseded by newer editions and are in order by date.


5.2.2 Fascicles

Volume 4's fascicles 0–4 were revised and published as Volume 4A.


6 In popular culture


7 See also

- *Introduction to Algorithms*

8 References

Citations

[1] The dedication was worded slightly differently in the first edition.

Notes


[6] TAOCP Future plans


Sources


9 External links

- Overview of topics (Knuth’s personal homepage)

- Oral history interview with Donald E. Knuth at Charles Babbage Institute, University of Minnesota, Minneapolis. Knuth discusses software patenting, structured programming, collaboration and his development of TeX. The oral history discusses the writing of The Art of Computer Programming.

- “Robert W Floyd, In Memoriam”, by Donald E. Knuth -(on the influence of Bob Floyd)

- TAOCP and its Influence of Computer Science (Softpanorama)
10 Text and image sources, contributors, and licenses

10.1 Text

- **The Art of Computer Programming** 

10.2 Images

- *File:KnuthAtOpenContentAlliance.jpg* 
  *Source:* https://upload.wikimedia.org/wikipedia/commons/4/4f/KnuthAtOpenContentAlliance.jpg
  *License:* CC BY-SA 2.5
  *Contributors:* Flickr.com (via en.wikipedia) *Original artist:* Flickr user Jacob Appelbaum, uploaded to en.wikipedia by users BeSherman, Duozmo

10.3 Content license

- Creative Commons Attribution-Share Alike 3.0